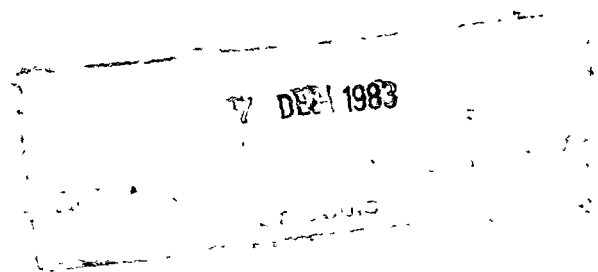


Numerical Simulations of Fuel Droplet Flows Using a Lagrangian Triangular Mesh

M. J. FRITTS, D. E. FYFE, AND E. S. ORAN

*Laboratory for Computational Physics
Naval Research Laboratory
Washington, DC 20375*

August 1983



Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center

(NASA-LEWIS # C-055049-D: NRL # 44-1422-0-1)

1 Report No NASA CR-168263		2 Government Accession No.		3. Recipient's Catalog No	
4 Title and Subtitle NUMERICAL SIMULATIONS OF FUEL DROPLET FLOWS USING A LAGRANGIAN TRIANGULAR MESH				5 Report Date August 1983	
				6 Performing Organization Code	
7. Author(s) M.J. Fritts, D.E. Fyfe, and E.S. Oran				8 Performing Organization Report No	
9 Performing Organization Name and Address Laboratory for Computational Physics, Code 4040 Naval Research Laboratory Washington, D.C. 20375				10. Work Unit No	
				11. Contract or Grant No. C-055049-D	
				13 Type of Report and Period Covered Contractor Report	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				14 Sponsoring Agency Code	
15 Supplementary Notes Final Report: Project Manager, C. J. Marek, Aerothermodynamics and Fuels Division, NASA Lewis Research Center, Cleveland, Ohio					
16. Abstract This report describes work done to convert the incompressible, Lagrangian, triangular grid code, SPLISH, to the study of flows in and around fuel droplets. This has involved developing, testing and incorporating new algorithms for surface tension and viscosity. The major features of the Lagrangian method and the new algorithms are described. Benchmarks of the new algorithms are given. Several calculations are presented for kerosene droplets in air. Finally, extensions which make the code compressible and three-dimensional are discussed.					
17 Key Words (Suggested by Author(s)) Droplets, Combustion Restructuring Mesh, Lagrangian			18. Distribution Statement Unclassified -- Unlimited		
19. Security Classif. (of this report) Unclassified		20 Security Classif (of this page) Unclassified		21. No. of Pages 44	
				22. Price*	

* For sale by the National Technical Information Service, Springfield, Virginia 22161

TABLE OF CONTENTS

	Page
SUMMARY.....	1
INTRODUCTION.....	2
LAGRANGIAN HYDRODYNAMICS ON A TRIANGULAR GRID.....	8
General Approach.....	8
Finite-Difference Algorithms.....	11
Adjusting and Restructuring the Mesh.....	13
NUMERICAL ALGORITHMS AND RESULTS.....	19
Inviscid Flow about a Droplet without Surface Tension.....	19
Surface Tension.....	22
Droplet Oscillations.....	25
Inviscid Flow about a Droplet with Surface Tension.....	26
Viscosity.....	29
Spreading of a Shear Layer.....	31
Flow about a Droplet with Viscosity and Surface Tension.....	32
FUTURE DEVELOPMENTS.....	35
CONCLUSIONS.....	38
ACKNOWLEDGEMENTS.....	40
APPENDIX.....	41
REFERENCES.....	43
FIGURE CAPTIONS.....	45

SUMMARY

In this report we describe the two-dimensional Lagrangian, incompressible Cartesian code, SPLISH, and the changes made to convert it for the study of flows in and around fuel droplets. The Lagrangian technique used in this study incorporates a general restructuring triangular mesh, which allows reconnection of vertices to eliminate grid distortions without adding numerical diffusion. This technique is accurate at material interfaces even though the interfaces undergo convolutions and may evolve into multi-connected surfaces.

New algorithms for surface tension and viscosity have been added to the basic fluid dynamics code. Surface tension is included as a jump in pressure across an interface by casting the surface tension forces in the form of a gradient of a potential. The surface tension algorithm is benchmarked by studying the oscillatory behavior of an $n = 2$ normal mode. The viscosity algorithm for a general mesh is presented and tested by calculating the spreading of a viscous shear layer.

We use the code to calculate the internal and external flows of oscillating and deforming kerosene droplets in an air jet. The surrounding air jet is initialized to laminar flow about a round kerosene droplet. The evolution of the droplet and jet are calculated from first principles, eliminating approximations for effective droplet size, wake effects or recirculation patterns. Results of the air-kerosene calculations are illustrated by sequences of frames from a computer generated movie of fluid particle positions. Both internal and external flows are shown as well as droplet distortion due to the relative flow. The algorithms needed to extend these calculations to compressible flows in three dimensions are discussed.

INTRODUCTION

Droplet combustion is a complex transient problem in multiphase flow. Particularly severe physical and mathematical approximations must be made to describe the detailed interactions between droplets and the external flow field in spray combustion models (Williams, 1973; Faeth, 1977,1983). For example, equivalent spheres are used to approximate droplet deformations, and empirical expressions are used to account for drag and convection. The effects of droplet breakup are included by using estimated breakup times and drop sizes after breakup. Quasi-steady flow approximations are used, and changes in the flow field due to droplet deformations, wake effects and droplet distortions due to the flow field are neglected. Finally, in most models the droplet concentration is assumed to be dilute since little is known about droplet-droplet or droplet-wake interactions.

The need for these approximations arises directly from the difficulty in following several physically distinct regions as they interact with the external flow field, distort and separate or merge. A Lagrangian technique is well suited to accurately modelling the transport of these various regions since it easily and naturally calculates the advection of boundaries. Because the various regions may severely distort and separate, the calculational grid must be able to self-consistently adapt to the physical flow. For these reasons the numerical technique used in this study is transient hydrodynamic modelling using a Lagrangian mesh.

The calculations are performed using the fully conservative, two-dimensional Lagrangian finite difference method developed by Fritts and Boris (1979) specifically to handle multi-phase flow. The method is based on a

dynamically restructuring Cartesian triangular grid. Triangle sides are aligned on material interfaces. Since vertex movement is Lagrangian, the interface sides accurately track the movement of the interface due to advection. A triangular grid avoids the problems of mesh tangling encountered in Lagrangian methods using a quadrilateral mesh: individual mesh points are continually reconnected to account for the migration of fluid elements in the flow field. Since the number of grid lines meeting at a vertex is variable, the resolution can be altered non-diffusively where needed (e.g., around a region of droplet distortion) without affecting the resolution in other areas of the computation. This is a major step forward in the computation of droplet flows because the Lagrangian technique allows for the evolution of the grid to multiply-connected regions. Thus, according to the flow conditions, droplets can break up and shatter.

Previously, the Lagrangian restructuring triangular grid technique has been applied to a number of incompressible fluid flow problems including calculations of nonlinear waves, flows over solid obstacles, Kelvin-Helmholtz and Rayleigh-Taylor instabilities, Couette flows and Taylor vortex flows (Fritts, 1976, 1976a, Fritts et. al., 1980, 1981, Emery et. al., 1981). This report details the research performed in adapting this technique to the study of droplet flows and presents calculations of the flow about droplets from first principles, without resort to models for droplet distortion, breakup, wake effects, drag or convection. The goals of the project are to extend this work further to build a comprehensive model for droplet combustion. This paper is a final report detailing the form of the model and indicating the results obtained in testing the model on purely hydrodynamic flows about droplets. Although the effects of combustion are not included in

this report, the model was constructed in such a way as to be able to account for the additional complexity introduced by combustion processes.

Consider the burning of an isolated heated droplet. The droplet heats up from its surface inward. Depending on the temperature, fuel, and other ambient conditions, the droplet may develop substantial internal gradients. If there is enough convection, recirculating flow develops within the droplet. As the surface evaporates, the locus of fuel and oxidizer at the fuel rich limit expands radially outward from the droplet. After the correct chemical induction time, the mixture may ignite outside this locus. For burning to continue, heat from the oxidizing gas and products must diffuse in past the products to continue the evaporation process, and fuel must continue to diffuse outward toward the flame region. Outside of the flame region the expanding oxidizing gas may sweep away unburned fuel vapor as well as combustion products. The Lagrangian technique improves upon phenomenological models because the advective motion within these several distinct regions are accurately calculated without prior assumptions of flow patterns and without approximating transient flows by steady-state or quasi steady-state flows. Adjustments of interface positions due to thermal, mass or energy fluxes across a triangle side may be calculated in a separate step using a conservative integral technique.

Although the physical processes described above are complex and highly nonlinear, the description is still idealized. A practical combustor must establish a relative velocity, V_d , between the fuel droplets and the hot, oxidizing gas. Unless V_d becomes substantial, surface tension keeps the droplet essentially spherical as it shrinks due to evaporation. For large V_d , the droplet distorts and may even shatter. As the droplets move, a

boundary layer of vaporized fuel and hot-gas develops which creates drag, decreases V_d and may introduce circulation within the droplet. Recirculation patterns may also develop outside the droplet, influencing both the boundary layer and external flows. The use of models to account for the droplet deformations and viscous effects may be avoided by incorporating algorithms for surface tension and viscosity into the hydrodynamics. This report describes these algorithms and the benchmarks used to validate their accuracy.

The shape of the flame surrounding the droplet depends strongly on the distorted external velocity field. As the Reynolds number increases, the flame shape around the droplet changes from an envelope to a wake flame. The site at which energy is released therefore changes, and this in turn readjusts the external flow field, local evaporation rates and species concentrations. To this picture must be added the interaction of burning and evaporating droplets. For close droplet spacings, the fuel-rich limit may extend around all the individual droplets and the resulting flame strongly resembles a diffusion flame. Large droplets may still migrate through the boundaries of the flame, and the combustion characteristics of these droplets may strongly affect the concentration of combustion products. Finally, the physical boundaries of the combustor may be important, altering flow patterns, temperature gradients, or other important properties of the system. The Lagrangian technique is well suited to tracking the individual or merging droplets, the transient flame shape and the flow by irregular combustor boundaries. Future additions to the technique will allow for compressible effects, heat release, evaporation and chemical reactions (Oran and Boris, 1981).

An adaptive grid technique can be implemented in two distinct ways: recomputing the entire grid at each timestep or locally restructuring the grid to eliminate distortions. Either method requires the storage of bookkeeping information necessary to compute the finite-difference templates. However, the mesh distortions which develop during a single timestep are confined to a fairly small number of triangles, typically at most 5 percent of the grid, so that the grid restructuring method can be an order of magnitude more efficient. This remains true even on vector machines, despite the fact that grid restructuring is inherently a scalar computation. The computer code used in this report was implemented on a TI-ASC parallel processor computer using vectorized scans to test for locales where grid restructuring might be necessary. Scalar routines then performed the actual calculations for grid restructuring in those regions only. The hydrodynamics routines were all vectorized where possible. Efficient coding for the ASC was obtained through Fortran-callable assembly language routines which optimized the scalar fetch and store operations. The main drawback of the technique is that the resultant programs are machine specific and must be recoded to run on other machines. Because the programs are not transportable, the code is not available for general distribution. Persons interested in obtaining program information or using the code should contact the authors directly.

The complete computer code is roughly 13,000 lines long, including documentation. Execution times on the ASC are typically about .01 seconds per timestep per grid point, including program diagnostics and output in the form of two three-color movie films, individual frames on fiche, and fiche listings. A movie supplement of various film sequences discussed in the text

is available for use with this report. Movies are generated through an optimized package which outputs to a Dicomed film recorder, Tektronix terminals or Calcomp plotters.

The calculations illustrated in the movie sequences and in the individual frames in the report all illustrate grids for a single droplet. Because the boundary conditions for all the calculations are periodic at the sides of the computational region and reflective at the top and bottom, the calculations represent an infinite series of droplets. However, most of the calculations made for this report terminate when the wake of the preceeding droplet impinges on the droplet following. This permits initial calculations of nearly isolated droplets. For the 125 micron drop size and the flow speeds used in the calculations, droplet distortions and possible shattering were expected, and single droplet simulations of this behavior could be compared more easily to experimental observations. The gridding routines can generate initial grids for any desired drop size within an arbitrarily large mesh. Large differences in resolution are permitted so that the droplet interface and interior can be well resolved despite larger grid sizes in some regions of the external flow field. The current code is therefore capable of simulations of arbitrarily large or small droplets.

1. General Approach

In principle, a Lagrangian formulation of the hydrodynamic equations is particularly attractive for droplet combustion calculations. Each fluid element is tracked as it evolves through the interaction with its changing environment and with external forces. Heat release, contaminant reactions and soot formation can all be represented locally, without resort to global models and without nonphysical diffusion. Conservation laws are simple to express since there are no fluxes out of the fluid element boundaries and the paths of the fluid elements themselves provide flow visualization. However, in all but the simplest flows the individual fluid elements deform, and these deformations are a severe hindrance to actually using a Lagrangian method.

In numerical calculations, fluid element distortion appears as stretching, shearing and eventual tangling of the computational grid. Although the use of a general-connectivity triangular mesh eliminates tangling, the accuracy of a calculation may still deteriorate when there are abrupt local changes in resolution and when the high-order effects of deformations are not represented. Therefore it is very important to pay close attention to how well conservation laws are satisfied. For example, the accuracy of the finite-difference algorithms for a general mesh may not be sufficient to conserve quantities advected with the fluid elements if some flux is allowed to flow out of elements to maintain straight lines in the computational grid. In the following section we show how exact conservation may be maintained.

The divergence and curl of the velocity field prescribe the kinetics of the field by specifying the local rate of expansion of the fluid, d , and local vorticity, $\bar{\xi}$, by

$$\begin{aligned}\nabla \cdot \bar{v} &= d \\ \nabla \times \bar{v} &= \bar{\xi} \cdot\end{aligned}\tag{1}$$

For incompressible flow, $d = 0$, and for irrotational flow, $\bar{\xi} = 0$.

For incompressible and irrotational flow in two dimensions, the velocity field is specified by a velocity potential ϕ and stream function ψ :

$$\begin{aligned}v_x &= \frac{\partial \phi}{\partial x} = \frac{\partial \psi}{\partial y} \\ v_y &= \frac{\partial \phi}{\partial y} = - \frac{\partial \psi}{\partial x} .\end{aligned}\tag{2}$$

These equations automatically satisfy the conservation of vorticity and mass, since

$$\nabla \cdot \bar{v} = \nabla \cdot (\nabla \times \psi) \equiv 0$$

(3)

and

$$\nabla \times \bar{v} = \nabla \times \nabla \phi \equiv 0 .$$

We would define finite-difference operators for divergence, curl and gradient which have these identical properties, and this requirement restricts the placement of variables. In particular, if ψ and ϕ are to be assigned to the Lagrangian vertices, the velocities \bar{v} must be specified at the centroids of triangles or the midpoints of line segments. Therefore the Lagrangian vertex velocities must be obtained by local averages.

For example, the first of Eqs. 3 will be recast in finite difference notation. The notation $\sum_{i(c)}$ is the sum over vertices i around a central vertex c . In such sums the sequence of vertices is assumed to be counter-clockwise around the central vertex. The quantity $A_{i+1/2}$

represents the area of the triangle having vertices $(c, i, i+1)$. In Figure 1 the area of triangle j is given by

$$2A_j = (\bar{r}_3 - \bar{r}_2) \times (\bar{r}_1 - \bar{r}_3) \cdot \hat{z}, \quad (4)$$

where \hat{z} is the direction out of the page. Similarly, for a scalar function f specified at each vertex and assumed piecewise linear within each triangle, the vector gradient of f (constant throughout the triangle j and discontinuous at the triangle sides) is given by

$$(\nabla f)_j = f_1 \frac{\hat{z} \times (\bar{r}_3 - \bar{r}_2)}{2A_j} + f_2 \frac{\hat{z} \times (\bar{r}_1 - \bar{r}_3)}{2A_j} + f_3 \frac{\hat{z} \times (\bar{r}_2 - \bar{r}_1)}{2A_j} \quad (5)$$

With this placement of variables the dynamics of the flow, as well as the kinematics, behave properly. That is,

$$\nabla \cdot \bar{v} = \nabla \cdot \nabla \phi = \nabla^2 \phi \quad (6)$$

is a general triangular grid Poisson equation which may be used to solve for the local pressure. At the same time, the pressures generated by forcing all local divergences to zero cannot by themselves alter the local vorticities, due to Eq.(3). In finite-difference form Equation 6 becomes

$$A_c \langle \nabla \cdot \bar{v} \rangle_c = \sum_{i(c)} \left\{ \phi_i \frac{\hat{z} \times (\bar{r}_c - \bar{r}_{i+1})}{2A_{i+1/2}} + \phi_{i+1} \frac{\hat{z} \times (\bar{r}_i - \bar{r}_c)}{2A_{i+1/2}} + \phi_c \frac{\hat{z} \times (\bar{r}_{i+1} - \bar{r}_i)}{2A_{i+1/2}} \right\} \times \frac{(\bar{r}_{i+1} - \bar{r}_i)}{2} \cdot \hat{z}, \quad (7)$$

where A_c is the area of the vertex cell, defined as one third of the sum of the areas of all triangles including that vertex.

The accuracy of the numerical algorithms is determined by both the local resolution and connectivity of the grid. For the approach used here, the local connectivity of the grid and the resolution are both determined in part by the requirements that the matrix generated from the Poisson equation, Eq.(7), remains diagonally dominant. With this restriction, convergence of an iterative solver for Eq.(7) is assured. The consequences of maintaining diagonal dominance will be given below in the discussion of grid restructuring algorithms.

2. Finite-Difference Algorithms.

In the previous section it was shown that as a result of specifying ψ and ϕ on vertices, the velocities must be specified at triangle centroids. With this definition the vorticity ζ about any vertex is readily calculated. Two formulations of the basic incompressible hydrodynamics equations are accessible with these definitions. For a ψ - ζ formulation the vorticity ζ is advanced at each cell for each timestep and the new stream function ψ is obtained from a solution of $\nabla \times (\nabla \times \psi) = \zeta$. By Eq.(3) the divergence of the velocity field is identically zero. Alternatively, in a P - \bar{v} formulation the changes in vorticity are zero by construction since $\nabla \times \nabla P = 0$. Then the new pressures are chosen to force the divergence of the velocity field to zero. The P - \bar{v} formulation has been the focus of previous work using this technique for several reasons. First, the ψ - ζ formulation becomes more complicated in three dimensions. Second, boundary conditions around bubbles and cavities are more complicated in the ψ - ζ version, particularly for droplet shattering and coalescence. Third, a self-consistent pressure field is usually desired even in the ψ - ζ formulation, requiring an extra solution step.

The P - \bar{v} algorithm specifies pressures, velocities and positions at full timesteps. A split-step algorithm is used to integrate the velocities forward half a time step, advance the grid a full time step, and then advance the velocities the remaining half time step.

$$\bar{v}_j^{1/2} = \bar{v}_j^0 - \frac{\delta t}{2\rho_j} (\nabla P)_j^0 - \frac{\delta t}{2} g\hat{y}, \quad (8)$$

$$\bar{v}_i^{1/2} = \frac{1}{2} (\bar{v}_i^0 + \bar{v}_i^n), \quad (9)$$

$$\vec{r}_i^n = \vec{r}_i^o + \delta t \vec{v}_i^{1/2}, \quad (10)$$

$$\vec{v}_j^{1/2} = \vec{R} \left(\{ \vec{r}_i^o \}, \{ \vec{r}_i^n \} \right) \cdot \vec{v}_j^{1/2}, \quad (11)$$

$$\vec{v}_j^n = \vec{v}_j^{1/2} - \frac{\delta t}{2\rho_j} (\nabla p)_j^n - \frac{\delta t}{2} g\hat{y}. \quad (12)$$

In these equations the subscript i denotes a vertex quantity, while the subscript j is used for triangle-centered quantities. Therefore the new vertex velocities appearing in the right-hand side of Eq.(9) are obtained from the area-weighted new triangle velocities found from Eq.(12) during the previous iteration. That is, Eqs.(8) and (12) advance the velocities according to the Lagrangian equations of motion: but since the grid is advanced at the half time step, a vertex velocity at the half time step must be found from the old and new triangle velocities. This implies an iteration over Eq.(9) through Eq.(12) to assure that the new velocities are indeed consistent with those used for the grid advancement in Eq.(10).

Equation (11) is the numerical expression of the change in the triangle velocities that must occur during the grid advancement if the vorticity is to remain constant for inviscid, homogeneous flow. This transformation is apparently a unique, but necessary, addition to Lagrangian methods to assure that vorticity is conserved. The actual form of the transformation has changed during this project, so further discussion will be deferred till later.

3. Adjusting and Restructuring the Mesh.

The primary advantage of a restructuring mesh is the flexibility which it permits for Lagrangian techniques in following long time solutions to complicated flows. Several types of local mesh adjustment and restructuring are used to maintain uniformity and accuracy of the discrete mesh representation. A mesh adjustment is a nonphysical movement or adjustment of the position of one or more vertices without changing the connectivity of mesh vertices. These adjustments are designed to regularize the mesh, and result in the effective transfer of fluid across triangle sides.

Mesh restructuring, on the other hand, does not generally involve movement of vertices but generally a redefinition of the mesh connectivity. Simplification of the mesh under restructuring may also involve vertex addition and deletion, but the positions of all other vertices remain unchanged. Therefore adjustment and restructuring are somewhat orthogonal procedures, one leaving vertex positions unchanged and the other leaving the mesh connectivity unchanged. Since restructuring always involves the changed position of a triangle side, it can also incorporate the nonphysical flow of fluid across triangle sides.

Both adjustment and restructuring represent departures from a purely Lagrangian description and threaten to introduce unwanted numerical diffusion into the method. To minimize diffusive and other errors, vertices and triangle sides lying on boundaries, surfaces and interfaces must be left undisturbed, and mass and momentum must be strictly conserved everywhere during both restructuring and adjustment. Although there are many schemes possible for mesh adjustment and restructuring, we have concentrated on a few

"primary" procedures from which more complex procedures may be developed. Since all conservation laws are satisfied for these simple procedures, all schemes built up from these primary procedures will also satisfy the same conservation laws.

A triangular mesh can quickly become distorted through the migration of vertices in the fluid flow, particularly for shear flows. This situation is typified by regions of long, narrow triangles bordering more regular ones. Without restructuring this distorted mesh forces the computation of derivatives using vertices which are no longer the nearest neighbors, and quickly leads to inaccuracies, numerical instabilities and nonphysical behavior. Time-step errors also become severe because of the disparity in size of triangle sides. For extremely distorted triangles, triangle inversion becomes likely. Because of the severity of these problems, grid restructuring must be imposed continuously to insure the accuracy of the numerical solution.

On a triangular grid, every nonboundary line uniquely specifies its two bordering triangles. These triangles form a quadrilateral for which the included line is one of the two possible diagonals. Figure 2a illustrates a configuration for which the present diagonal (solid line) should be reconnected to the opposing diagonal (dashed line). One possible algorithm always chooses the shorter diagonal unless reconnection produces too large a disparity in triangle areas. This safeguards against reconnecting the diagonal of inverted quadrilaterals to produce a negative area triangle, as shown in Fig. 2b.

The reconnection algorithm could instead be formulated to ensure diagonal dominance of the triangular grid Poisson equation (Eq.7) as mentioned

previously. Note that the coefficient of the ϕ_c term in Eq. 7 is

$$a_c = - \sum_{i(c)} \frac{|\bar{r}_{i+1} - \bar{r}_i|^2}{4A_{i+1/2}}, \quad (13)$$

and is always negative. The coefficient a_i of the ϕ_i term is

$$a_i = - \frac{(\bar{r}_c - \bar{r}_{i+1}) \cdot (\bar{r}_{i+1} - \bar{r}_i)}{4A_{i+1/2}} - \frac{(\bar{r}_{i-1} - \bar{r}_c) \cdot (\bar{r}_i - \bar{r}_{i-1})}{4A_{i-1/2}}. \quad (14)$$

Equation 14 reduces to

$$a_i = \frac{1}{2}(\cot \delta_{i+1/2} + \cot \delta_{i-1/2}), \quad (15)$$

where $\delta_{i+1/2}$ and $\delta_{i-1/2}$ are the angles in the $(i+1/2)$ th and $(i-1/2)$ th triangles opposite the line from c to i . If the sum of $\delta_{i+1/2}$ and $\delta_{i-1/2}$ is less than π radians for each i , the matrix is diagonally dominant and normal iterative procedures for inverting Eq.7 work well. If for any i

$$\delta_{i+1/2} + \delta_{i-1/2} > \pi \text{ radians} \quad (16)$$

then the line from c to i is reconnected to join $(i+1)$ to $(i-1)$. This procedure therefore chooses the other diagonal of the quadrilateral formed by the $(i+1/2)$ th and $(i-1/2)$ th triangles. Since the sum of angles in a quadrilateral is just 2π radians, then the new diagonal has opposite angles that sum to less than π radians. The new matrix coefficients generated by this connectivity again ensure that the matrix is diagonally dominant. Note that negative area triangles cannot form with an algorithm that requires that the sum of the opposing angles is greater than zero and less than π radians. For all simulations presented in this paper, the algorithm enforcing diagonal dominance was used.

The reconnection algorithm is complicated by the need to uphold conservation laws. To conserve vorticity locally, the new triangles defined by a reconnection have velocities constrained to those which leave the

vorticity about each vertex unchanged. The additional requirement that the momentum is locally conserved uniquely specifies the post-reconnection velocities for the two new triangles. The algorithm resulting from these constraints is reversible. Replacing the reconnected diagonal with the original diagonal redefines the initial two triangles with their identical original velocities.

A further complication to the reconnection algorithm arises at material interfaces. Since triangle sides aligned along interfaces cannot be reconnected, diagonal dominance cannot be preserved for matrix coefficients from interface vertices by reconnection. Alternatively, a vertex may be added at the midpoint of the interface line so that the opposing angles are bisected by the lines drawn from the new vertex to the opposite vertices. This scheme assures diagonal dominance while increasing the resolution in the neighborhood of the interface. This algorithm is exercised when vertices close to the interface move toward the interface or when the interface becomes severely deformed. In either case, more resolution at the interface is generally required.

The interface problem indicates that reconnection cannot solve all the mesh readjustment problems encountered in complex fluid flows. Two additional "primary" procedures are required: vertex addition and deletion. As discussed above, the addition of a vertex on an existing interface line is accompanied by the insertion of two new lines to form two new triangles. For a line on the boundary only one new line and triangle are added. The new vertex may be added anywhere along any interior, interface or boundary line, since later reconnections can be used to restore diagonal dominance. Two new triangle velocities must be specified, and these are selected in accordance

with the same conservation laws used for the grid restructuring algorithms.

Vertices may also be added in the interior of any single triangle. Simple algorithms for vertex addition within a triangle typically leave the grid motion unchanged. The three new triangles are circumscribed by the original triangle whose motion is usually constrained just as if the new vertex was not there, since the divergences, and subsequent pressure changes, are the same. To be effective, addition of a vertex within a triangle must be accompanied by the reconnection of at least one of the triangle sides. Physical variables centered at the vertex are found by interpolation just as in the case of addition on a line.

Deletion of a vertex is performed by the inverses of these two processes. To delete a point within the interior of a subregion, reconnections are first made to isolate the point within a triangle. The vertex, three lines and two triangles are deleted, and the new physical variables are determined by averages over the old configuration. Subsequent reconnections enforce diagonal dominance. To delete a point on an interface, the interface must first be realigned to its new, lower resolution position either by unphysical motion of the interface vertex or by changing the physical properties centered at one of the triangles. The inverse process can then be used to eliminate the vertex and redefine physical variables in accordance with the conservation laws. It should be noted that the reconnection algorithm may require the addition of a vertex at an interface for diagonal dominance, and may violate a resolution requirement stipulated in another part of the set of grid restructuring algorithms. Such conflicts can cause cycling through the addition and deletion algorithms unless the requirements are properly tailored to both requirements.

General grid restructuring algorithms are built from these primary functions, each of which is incorporated into a Fortran subroutine. The more general routines for grid restructuring are used to provide flexibility in setting resolution requirements and eliminating conflicts among the lower level functions. Input to these routines is in the form of a user specified resolution requirement limiting the maximum and minimum size for triangles and line segments and a CFL parameter which determines the time step from the flow speed. For the calculations presented in this paper these specifications were global in nature, although local specifications could be used which would be based, for example, on material type, distance from interfaces or gradients in physical properties. Because the global resolution was determined by a range of acceptable sizes with the maximum near the initial values, finer resolution will persist wherever grid restructuring has occurred, as is evident in the figures below illustrating computational grids for several different problems. Most of the finer resolution arises near interfaces, where the reconnection routines force the addition of vertices to ensure diagonal dominance and sufficient accuracy. Local resolution specifications were used only in the grid initialization algorithm, which is built from the same primary routines.

NUMERICAL ALGORITHMS AND RESULTS

The basic two-dimensional hydrodynamics computer code was constructed in such a way that the finite-difference operators for divergence, curl and gradient exactly reflected the properties of the continuum operators. This construction assures conservation of vorticity and mass and provides a basis for determining the local grid connectivity. The extensions to this code described below are all made in exactly the same spirit: the finite-difference approximations to both physical and dynamical processes conform to the continuum limit and conserve properly.

The expansion of the basic triangular mesh code to droplet flows was programmed to occur in several stages. Each stage involved the development of new algorithms for particular additions to the physics being modelled or for necessary new numerical techniques, and the benchmarking of these new algorithms against relevant physical calculations. The following sections detail the progress achieved at the separate stages, the algorithms developed and the numerical results of the benchmark calculations.

1. Incompressible, Inviscid Flow about a Droplet without Surface Tension

The first test problem for the code was a simulation of incompressible, inviscid flow about a cylindrical droplet with a density twice that of the background fluid. Gridding routines were written to position an arbitrarily large drop at the center of the computational grid for variable resolution inside and outside the droplet. Additional routines initialize the flow by a pressure pulse at the left boundary for the first half timestep or ramp up

the flow speed if the initial perturbation would cause adverse transients. Divergence calculations for advancing the pressure during the ramp-up time correct for the effect of the initial pressure pulse. For all later times the left and right boundary condition is periodic. Rigid wall boundary conditions are imposed at the top and bottom of the computational region for all times. The modeled problem is therefore a row of droplets in an impulsively started flow field. The droplet is gridded into 28 triangular computational cells in a total system of 552 cells, as seen in the first frame of Fig. 3. Only the triangle vertices are shown within the droplet.

Figure 3 shows the triangular mesh at several times in the calculation. Pathlines of each of the vertices are plotted as a diagnostic, but are not included in this Figure. Early in the calculation a recirculation zone forms behind the droplet, compressing the droplet in the direction parallel to the flow. Flow within the droplet is initiated by this compression in a direction normal to the external flow. The bulges formed at the top and bottom of the distorted droplet are pulled around the recirculation zone by the shear flow which is at a maximum at these points. The internal droplet flow is therefore driven by the compression set up between the front and rear stagnation points and by the high shear flow which extends around the top and bottom of the droplet and recirculation zone. The interaction of the droplet back onto the external flow occurs primarily through the enlarged cross-sectional area presented by the droplet to the flow, which increases the size of the recirculation zone. Eventually, as seen in Figure 3, the droplet is squeezed into a thin layer coating the recirculation zone. The thinned film then shatters into several smaller pieces, first at the rear of the droplet and later in the more laminar flow toward the front of the droplet.

A study of the pathlines of the Lagrangian particles shows that the flow is regular at all times despite the distorted shape of the droplet. Subtracting off the mean flow from the calculated flow field would show a large stationary recirculating double vortex. In a spherical geometry this recirculation zone would form a vortex ring, and the thinned droplet coating the ring would fragment in both the radial and azimuthal directions. The varying density of triangle vertices arises because higher resolution is required in the vicinity of the droplet interface.

A second calculation was performed for a 10:1 ratio of droplet to external fluid density. The initial flow is quite similar and shows back flow at the rear stagnation point as well as internal droplet flow normal to the external flow. However, later droplet development is substantially altered. The more massive droplet is less easily deformed about the recirculation zone, and as a result the droplet grows more than the 2:1 case in the direction normal to the external flow. Therefore a more symmetrical front-to-back flow pattern develops. With no surface tension, there is no restoring force and no steady-state shape. The droplet grows normal to the flow until it is thinned sufficiently to break. Edge effects due to the proximity of the top and bottom of the computational region are clearly visible by the end of the calculation.

The results of both tests agree qualitatively with existing theory and experiment. Because of the lack of surface tension, no quantitative comparisons could be made. The gridding algorithms were found to be sufficient to represent the droplet down to the desired resolution as input to the calculation. For both calculations the grid adjusted itself automatically, i.e. without need for user intervention, to the changing

connectivity of the shattering droplet and to the flow about the stagnation points. The Lagrangian pathline diagnostic was found to be effective in illustrating recirculating flow in a movie format. During the debugging phase of these calculations it was also necessary to develop a new diagnostic routine to zoom in and plot the grid and local field variables at points where the code indicated problems at gridding anomalies or in convergence of the Poisson solver.

2. Surface Tension.

The incorporation of surface tension into the code was a learning process in finite differencing over distorted grids. Surface tension is conventionally cast into a finite-difference form by fitting vertices on the material interface to some parametric function from which an estimate of local curvature can be made. Once the curvature is known, a surface tension force is evaluated and used to accelerate interface vertices. This scheme fails for two reasons. First, the interface vertices are accelerated directly by surface tension forces evaluated on the vertices. Since velocities are centered on triangles in SPLISH, unless a secondary calculation is made, the velocity field sees the effect of the acceleration a half-timestep later, and as a result the pressure calculated within the droplet is inconsistent with that found from the surface tension formula. Secondly, since the pressure gradient forces and surface tension forces are not calculated in the same manner, numerical error results which grows with each timestep.

Both of the problems mentioned above were eliminated by a new and unique formulation of surface tension in which a surface tension potential is used

to generate the forces. Since the pressure gradient forces are calculated in the same manner and on the same grid as those derived from the surface tension potential, exact balance can be achieved between the forces and static pressure drops across the interface agree exactly with theory. The surface tension force is then formulated as a gradient of a potential present only at the surfaces. Therefore both the "surface tension potential" and the pressure are dynamically similar, and the physical pressure drop across the interface must exactly cancel the surface tension forces. Since the surface tension is normal to the interface and opposes the pressure drop (Fritts, et. al., 1982), then the $\nabla p \times \nabla p$ terms which alter the vorticity are zero for the finite-difference algorithms.

The finite-difference algorithms for surface tension are therefore quite simple in form. The surface tension forces are included through Laplace's formula for the pressure jump across an interface (Landau and Lifshitz, 1975)

$$P_i - P_o = \sigma/R \quad (17)$$

where P_i is the pressure just inside the droplet at the interface, P_o is the pressure just outside the droplet at the interface, σ is the surface tension coefficient associated with the two media which define the interface, and R is the radius of curvature of the cylindrical droplet. The radius of curvature is positive at points on the interface where the droplet surface is convex (a spherical droplet is convex everywhere) and negative when the droplet surface is concave. These pressure jumps are included in the Poisson equation for the pressure. The average pressure, $(P_i + P_o)/2$, is computed at an interface vertex. From the average pressure and the pressure jump we can compute a pressure gradient centered on triangles, within and

without the droplet, for inclusion in the momentum equation.

The radius of curvature is computed from a parametric cubic spline interpolant to the interface vertices. If we denote the interface vertices by $\bar{r}_i = (x_i, y_i)$, $i=1, \dots, N$, with $\bar{r}_N = \bar{r}_1$, we define a pseudo arc length parameter, s , so that the spline knots occur at the points

$$\begin{aligned} s_1 &= 0 \\ s_i &= s_{i-1} + |\bar{r}_i - \bar{r}_{i-1}| \quad i=2, \dots, N. \end{aligned} \quad (18)$$

We then generate the twice differentiable periodic spline interpolants $x(s)$ and $y(s)$ from the data s_i, x_i, y_i , $i=1, \dots, N$ as prescribed by deBoor (1978). The curvature is then given by

$$\left| \frac{1}{R} \right| = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{3/2}} \quad (19)$$

where a prime indicates differentiation with respect to the parameter s . The sign of R at an interface vertex, \bar{r}_i , is given by the sign of

$[(\bar{r}_{i+1} - \bar{r}_i) \times (\bar{r}_i - \bar{r}_{i-1})] \cdot \hat{z}$, where \hat{z} is the unit vector in the z direction.

The parametric spline fit is also used for regridding. When the regridding algorithm calls for the bisection of a triangle side which borders the two media, a new vertex is added on the spline interpolant between the indicated vertices rather than bisecting the straight line segment. A straight line bisection introduces spurious interface oscillations (Foote, 1973) whereas bisecting the spline maintains the general overall shape of the interface.

a. Droplet Oscillations.

In order to test our algorithm for surface tension, we performed calculations of droplets which oscillate under the effects of surface tension. A linear theory for small amplitude oscillations on cylindrical jets was first given by Rayleigh (1879). When a perturbation is totally in the plane perpendicular to the axis of the cylinder, Rayleigh found the frequency, ω , for the oscillations is given by

$$\omega^2 = (n^3 - n) \frac{\sigma}{\rho a^3} \quad (20)$$

where the surface of the jet is given in polar coordinates by

$$r = a + \epsilon \cos(n\theta). \quad (21)$$

From Equation (20) it is clear that the lowest oscillating mode is given by $n=2$. Rayleigh used Equation (20) to interpret his experiments with jets. For large amplitude oscillations he found the experimental frequency to diverge from that predicated by his linear theory and attributed errors to nonlinear effects.

In the numerical calculation presented we study an $n = 2$ oscillation. We have taken the parameters $a = 0.0125$ cm, and $\sigma = 30$ dynes/cm, values which are typical for droplet combustion problems. We used a droplet density of 2g/cm^3 and an external fluid density of 1g/cm^3 . The results of a calculation with $\epsilon = 0.2a = 0.0025$ cm are shown in Figure 4. The numerical oscillation period is approximately 1.25×10^{-3} s. In order to compare this result with Rayleigh's theory, we must first correct his result for the effect of the presence of the external fluid. Equation (20) then becomes

$$\omega^2 = (n^3 - n) \frac{\sigma}{(\rho_d + \rho_e) a^3} \quad (20')$$

where ρ_d is the droplet density and ρ_e is the density of the external fluid. With the period defined as $2\pi/\omega$, Equation (20') gives a period of 1.13×10^{-3} s. The discrepancy between the numerical and theoretical results can be explained by the finite grid spacing. However, given Rayleigh's experience with large amplitude oscillations, it is reasonable to expect our computational period to differ somewhat from that given by the linear theory. Further calculations were performed with smaller amplitudes, ϵ , to see if any of the difference is attributable to nonlinear perturbation effects or if the linear theory is directly applicable in this regime. In addition, different density ratios were used, viz. 10:1 and 800:1, which more closely approximated a combustion environment and which allowed the testing of the effects of the external fluid density on the numerical convergence of the pressure algorithm. The net result was that all the difference between theory and the numerical result is consistent with second-order convergence to the theoretical frequency for small perturbations and small grid size.

b. Incompressible, Inviscid Flow about a Droplet with Surface Tension

The second test of the surface tension algorithm was a recalculation of the initial benchmark problem, but with surface tension forces turned on. This test was necessary to check whether the code could allow for more radical interface deformations and whether the spline fit would properly allow the droplet to separate into smaller droplets or, alternatively, for many smaller droplets to coalesce.

Figure 5 shows the results of a calculation with surface tension for the same initial conditions as used in the calculation without surface tension (Figure 3). As in the case without surface tension, the internal droplet flow is driven by compression parallel to the external flow and is initially normal to the external flow. A recirculation zone is formed in the wake of the compressed droplet and the droplet deforms into a kidney shape between the opposing streams of the external flow and the recirculation zone. With the relatively large surface tension forces used in this calculation, further stretching of the droplet is curtailed. Instead of the droplet deforming into a film around the recirculation zone, the rear of the droplet begins to oscillate under the restoring force provided by surface tension. The oscillation arises at the rear of the droplet at a wavelength equal to the droplet diameter. The large deformations seen at later times have the shortest wavelengths which can be supported by the grid resolution at those times. These higher modes are excited numerically through wiggles induced by the spline fit to the interface vertices and by physical oscillations induced by the recirculating flow at the rear of the droplet. The spline routines have been recoded for a higher order spline, but these algorithms have not been incorporated into the main routines at the time of this report. The front of the droplet remains smooth throughout the calculation despite the large nonlinear oscillations occurring at the rear of the droplet, and the general droplet shape and behavior are consistent with experiments performed in low viscosity fluids.

The Lagrangian pathlines for the vertices again show the development of a recirculation zone in the wake of the droplet. Initially this zone is similar to the one in the calculation without surface tension. The primary

effect of surface tension on the external flow is that oscillations are superimposed on the recirculating flow at the rear of the droplet. For the calculation shown here, this oscillation is sufficient to disrupt the regular flow pattern in the droplet wake and to induce higher mode oscillations. The effect of surface tension on the internal droplet flow appears in the retardation and cessation of droplet thinning around the recirculation zone and in the increased mixing due to the droplet oscillations. The internal flow remains laminar at the front of the droplet even in the presence of the large oscillations at the rear. The external and internal flow patterns and droplet shape at later times agree qualitatively with experimental shapes and flow patterns at high Reynolds number (Clift et. al., 1978). This agreement extends to three dimensional droplets as well since experiments of bubbles and droplets between parallel plates show results similar to experiments of unconfined droplets and bubbles at their planes of symmetry. The calculation does not include viscosity so that the Reynolds number is large and limited only by an effective cell Reynolds number.

3. Viscosity

The next step in the construction of the droplet combustion model is to include algorithms for viscosity and compressibility. The equations of motion of a viscous fluid in two dimensions require the additional terms

$$\frac{dv_x}{dt} = \dots + \nabla \cdot v \nabla v_x \quad (22)$$

$$\frac{dv_y}{dt} = \dots + \nabla \cdot v \nabla v_y$$

As discussed above, the formulation of the finite-difference algorithms required velocities to be specified at triangle centroids. Gradients of velocity components such as those in Equation (22) are therefore difficult to express to high order accuracy and the regions over which the approximations must be made are irregular and costly to compute.

A numerical algorithm which is easier to implement can be derived by expressing the change in vorticity at a grid point due to viscosity as

$$\frac{d\xi}{dt} = \nu \nabla^2 \xi \quad (23)$$

since in two dimensions the vorticity is in the z direction. The easiest algorithm to implement is one which introduces the necessary changes in the triangle velocities about a vertex to satisfy Equation (23). If the choice is made to enforce equal contributions from each of the triangles about the vertex, then Equation (22) is satisfied. For example, consider an unperturbed shear flow parallel to the x -axis. For this flow ∇v_x defines

the shear profile and ∇v_y is zero. Therefore by Equation (22) only the x-components of velocity in regions adjacent to the shear layer should change and the y-components of velocity should remain unchanged. The choice of equal triangle contributions to the change in vorticity dictated by Equation (23) ensures that the numerical algorithm will induce change only in the x-velocities while the changes in y-velocities will be identically zero. In this sense, Equation (22) is used as a conservation law to ensure proper behavior of the finite difference algorithm.

Although this algorithm was simpler to code, the specification of equal contributions from all triangles about a vertex was difficult to enforce except for regular grids. The determination of how the required changes in vorticity were to be translated to velocity changes was ambiguous for different grid geometries. The algorithm produced the correct spreading rates for a shear profile, but only for very regular grid geometries. For an arbitrary grid a more detailed prescription was necessary.

A discretization for which $\nabla^2 \bar{v}$ is a triangle-centered quantity as in Equation (22) remains desirable. If in the finite-difference formulation for Equation (22) the coefficient of viscosity is centered on triangles, any ambiguity at interfaces is avoided for stratified fluids, whereas special algorithms would be needed for a vertex-centered coefficient of viscosity. This placement of variables puts the viscosity on the same footing as the density. Temporal changes in the triangle velocities are straightforward to compute, since now

$$\frac{d\bar{v}}{dt} \sim v_t (\nabla^2 \bar{v})_t , \quad (24)$$

where the subscript "t" indicates that all quantities are triangle centered.

a. Spreading of a Shear Layer.

This algorithm was tested in a calculation of the spreading of a shear layer of initially zero thickness given by

$$\bar{V}(x, y, t=0) = \pm V_x \text{ for } y \begin{matrix} > \\ < \end{matrix} y_0 ,$$

where y_0 is the original location of the vortex sheet. The velocity distribution across this layer will evolve as

$$\bar{V}(x, y, t) = V_x \operatorname{erf} \left(\frac{y}{(4\nu t)^{1/2}} \right) \quad (25)$$

and the width ΔY of the layer will grow as

$$\Delta Y \sim 8(\nu t)^{1/2}.$$

For the test calculation the grid was initialized to center a vortex sheet in a grid 16 cells wide with an initial layer width of zero. The two opposing streams had initially constant velocity profiles and the evolution of the interface between the streams was governed by the same algorithms as the interior of either fluid with no special interface boundary condition. The boundary condition at the sides of the computational region were periodic, and the top and bottom had free-slip boundary conditions. At the end of the calculation the layer width agreed exactly with the theory and the layer extended over the whole mesh. The velocity profile for each stream coincided with that given by Equation (25) to within round-off error. The y-components of the velocity remained zero indicating that the algorithm was working well for the grid distortions presented by the problem.

b. Incompressible Flow about a Droplet with Viscosity and Surface Tension.

Before the droplet runs began two modifications were made in the code. The first was the addition of a new initial condition. All previous runs had used an impulsively started air flow. With the addition of viscosity, this led to large momentum transfers across the droplet interface early in the calculation. The new initial condition specifies a steady-state flow field derived from a streamfunction calculation. This provides a much smoother start and a closer representation of the actual physical conditions the droplet would see. The second modification was to the residual error algorithm which corrects for the effects of keeping straight triangle sides. A mistake was found which became evident only for large momentum transfers across an interface. The error was corrected and the problem was eliminated.

The first viscous calculations were of air flow past a kerosene droplet and included the effects of both surface tension and viscosity. The physical parameters were appropriate for a combustor environment:

density of kerosene	0.82 g/cm ³
density of air	.0013 g/cm ³
surface tension (STP)	30 dynes/cm
viscosity of kerosene	1.8 centipoise
viscosity of air	.018 centipoise
air velocity	100 m/sec
droplet radius	125 microns.

Figure 6 follows the evolution of the internal and external flow fields for the calculation. At an air velocity of 100 m/sec and a droplet radius of 125

microns, the corresponding Reynolds number is roughly 1600. Boundary conditions for the computation are periodic on the sides of the computational region and reflective at the top and bottom. The passage of fluid through the mesh can be tracked by the pathlines of the uppermost and lowermost vertices next to the top and bottom of the computational region. These vertices are slightly above and below their neighbors due to the algorithm used to calculate the initial grid. Their position can be tracked through all nine frames, showing that the fluid originally behind the droplet has progressed through the mesh and has interacted with the face of the (next) droplet. Note the initial frame is not at $t=0.0$ in order to accumulate particle pathlines which are indicative of the originally laminar flow.

The first clear indication of the development of the recirculation region is seen in the fourth insert where a pair of counter-rotating vortices are evident. The recirculation zone continues to develop throughout the calculation, although at times the vortex pair is not as evident due to the deletion and addition of vertices which interrupts the continuity of the pathlines. By the last insert it appears that another pair of vortices is forming near the droplet, indicating that the original pair may be shed. There is now large distortion of the leading face of the droplet, and the droplet is about to enter the wake of the preceding droplet. Distortions in the face of the droplet are evident by at least the seventh frame, and appear to be due to increased curvature and condensing of the streamlines in the external flow caused by the approaching wake. The internal velocities are small compared to the external flow rates and therefore cannot be distinguished as pathlines. However, an indication of the (small) internal recirculation may be obtained by comparing internal vertex positions at various

timesteps. Figure 7 shows the grid at the corresponding times in the calculation. Vertex addition has evidently occurred primarily where needed, in the developing wake of the droplet and all along the droplet interface.

Figure 8 shows the pathlines for a simulation with identical parameters except for the flow speed, which is increased to 120 m/sec for a Reynolds number of 2000. The fluid now completely passes through the mesh, with the fluid initially near the droplet completely passing the next droplet. Therefore the droplet has interacted with the wake of the preceding droplet for one droplet diameter. The initial flow about the droplet is seen to be quite similar except for a more pronounced flattening of the face of the droplet due to the higher flow speed. The wake develops in much the same manner, but it now interacts strongly with the flow at the forward stagnation point on the droplet. Oscillations in the flow due to the wake are transmitted to the forward face of the droplet and give rise to fairly large perturbations. As seen in Figure 9, the computational grid is in need of further refinement at this time because the perturbations cannot be resolved by the length scales originally chosen for the run. One of the crests of the surface wave is gridded by a single triangle, a situation which allows no communication of that surface fluid with the interior of the droplet. In order to continue the simulation better resolution must be obtained about the droplet surface. A new algorithm has been developed to permit higher resolution near points of large curvature of material interfaces, but the algorithm was not implemented at the time of these calculations.

FUTURE DEVELOPMENTS

The next step in the construction of the droplet combustion model is to include an algorithm for compressibility. The addition of compressibility will occur in two ways depending on the characteristic flow velocities in the calculations. When flow speeds are slow with respect to the sound speed, we do not want the timestep to be limited by the Courant condition. In such cases the sound waves can be filtered out of the solution by altering the pressure field to account for local divergences on the time scale of the fluid flow (Jones and Boris, 1979). These divergences, which arise for example, from heat release, are introduced into the pressure Poisson Equation in a manner similar to that for incompressible flow. However, there is a restriction that the relaxation times occur at the proper time scale. For the triangular mesh, such additions should be easy to implement since a divergence correction term is already used to account for the effects of maintaining straight triangle sides.

In the case for which sound waves must be included, an energy evolution equation and an equation of state must be included in the finite difference algorithms as well. The algorithm which will be used for the equation of state expresses the density as a function of the pressure and energy. Given a new internal energy derived from the energy evolution equation and an approximation to the pressure, density is calculated from the equation of state. This equation of state density is compared to the density derived from the fluid dynamics and the difference is iterated to zero. This solution is the inverse of the usual algorithms for the equation of state which express pressure as a function of density and energy. The method has

been tested extensively for a one-dimensional restructuring mesh in the code ADINC (Boris, 1979; Fritts et al., 1981). The technique works well in one dimension and exhibits diminished finite difference error propagation due to the fact that numerical errors in pressure result in small density fluctuations. In the usual technique, small density errors can give rise to large pressure fluctuations, and hence a larger error propagation.

An energy evolution equation,

$$\frac{dE}{dt} = - E \nabla \cdot \bar{\mathbf{v}} - \nabla \cdot (P\bar{\mathbf{v}}) + \nabla \cdot \lambda \nabla T, \quad (26)$$

will also be needed to account for the effects of thermal conductivity, represented by the last term in Eq.(26). If both energy and temperature are carried as vertex centered quantities, then there are no new techniques required for the first and third terms, since the finite-difference approximations for similar terms are well tested. The center term must be carried as an average, since pressures are vertex centered while velocities are triangle centered. The incorporation of reactions is rather straightforward and will follow previously tested techniques given by Oran and Boris (1981).

The three-dimensional analogue of a triangular grid is a tetrahedral grid in which surfaces are tessalated by triangles. Although the addition of one more dimension introduces new complications in the reconnection algorithms, much of what was learned from the two-dimensional case carries over intact into three dimensions. Vertices can still be deleted by successive reconnections to isolate a vertex within a single tetrahedron. At that point, the vertex and its four lines can be eliminated from the grid. Vertices can be added within tetrahedra, in the plane of a triangle and on lines without major modification of the techniques used in two dimensions.

The conservation criteria used for reconnecting, adding and deleting cells in three dimensions usually involves either extending integrals to one higher dimension, or measuring an angle between planes rather than lines.

Similarly, the hydrodynamics finite-difference algorithms are logical extensions of the two-dimensional algorithms. The use of primitive variables allows a simple extension for the vorticity integrals, and the solution of Poisson's Equation still requires just one matrix inversion.

CONCLUSION

In this report we have described the basic algorithms in the two-dimensional Lagrangian, incompressible Cartesian code, SPLISH. A main advantage of the Lagrangian technique is the general property of the restructuring triangular mesh, which allows reconnection of vertices without adding numerical diffusion. This technique is accurate at material interfaces even though the interfaces undergo convolutions and may evolve into multi-connected surfaces. Because of the potential advantages of such a technique to combustion problems, we have begun the process of converting the code for the study of flows in and around burning fuel droplets.

We have described and presented benchmarks for two new algorithms which have been added to the basic fluid dynamics code: one for surface tension and one for viscosity. Surface tension is included as a jump in pressure across an interface by casting the surface tension forces in the form of a gradient of a potential. The algorithm has been benchmarked by comparing numerical solutions of the oscillations of an $n = 2$ normal mode to the results of an analytic solution. The difference between the exact and numerical solution becomes smaller as grid resolution is improved. The viscosity algorithm is presented and tested by calculating the spreading of a viscous shear layer and comparing this to the analytic solution. Here the analytic and numerical solutions are virtually identical.

Finally, we have discussed initial calculations of the behavior of dense fuel droplets in a flowing gas. Droplet flows with and without surface tension and with and without viscosity are discussed. Calculations of kerosene droplets in air are presented. These show both internal and

external droplet flows as well as distortion of the droplet due to the relative flow. Also, we see how vortex pairs develop and are shed behind the droplet. Droplet-droplet interactions occur when the distorted flow induced from one droplet reaches another. The results of the calculations are illustrated by sequences of frames from a computer generated movie of fluid particle positions.

The restructuring mesh has been shown to be capable of accurately tracking interfaces despite transition to multiply connected flows. As a result calculations of distorting and shattering droplets can now be performed entirely from first principles, without recourse to approximations or phenomenological models. The numerical technique is appropriate to the tracking of flame fronts as well. There are a number of future directions that can be taken in the development of the model. Algorithms which make the code compressible have been developed and must be implemented. When this is done, we can consider effects such as thermal conduction and chemical reactions. Implementation of these algorithms in the cylindrical version of this code instead of the currently used cartesian version would allow the study of a round instead of cylindrical droplets. Some of the basic algorithms need to extend the code to three-dimensions have been worked out with tetrahedra replacing triangles. However, these need considerable development before they can be used here.

ACKNOWLEDGEMENTS

This work has been sponsored by the National Aeronautics and Space Administration and the Naval Research Laboratory. Support for prior research, which formed the basis for this work, was provided by the Office of Naval Research and the Naval Research Laboratory.

APPENDIX

Presentations and Papers Supported by C-055049-D.

- 1) Numerical Simulations of Flames and Detonations: Current Methods and Capabilities, E. S. Oran, J. P. Boris, M. J. Fritts, M. Fry, D. Fyfe, K. Kailasanath, M. Picone, J. Tittsworth and T. Young, International Specialists Meeting on Fuel Air Explosions, Montreal, Canada, November 4-6, 1981.
- 2) Numerical Calculations of Flow in and Around a Droplet, D. E. Fyfe, M. J. Fritts, and E. S. Oran, Bull. Am. Phys. Soc. 26, 1265 (1981).
- 3) Calculations of Flow in and About Fuel Droplets, M. J. Fritts, E. S. Oran and D. E. Fyfe, Eastern States Meeting of the Combustion Institute, October, 1981.
- 4) Numerical Simulations of Droplet Flows with Surface Tension, M.J. Fritts, D.E. Fyfe, and E.S. Oran, Proc. of the ASME meeting, 82-WA/HT-17, 1982.
- 5) Numerical Simulation of the Behavior of Fuel Droplets, M. Fritts, D.E. Fyfe, and E. Oran, eastern Section: The Combustion Institute, Atlantic City, December, 1982
- 6) Effects of Surface Tension and Viscosity on Flows in and around a Droplet, M.J. Fritts, D.E. Fyfe, and E.S. Oran, American Physical Society, Division of Fluid Dynamics, New Brunswick, Nov 1982
- 7) Numerical Simulations of Fuel Droplet Flows, M.J. Fritts, D.E. Fyfe, and E.S. Oran, submitted to the Proceedings of the Ninth International Colloquium on the Dynamics of Explosive and Reactive Systems, Poitiers, France, July, 1983.

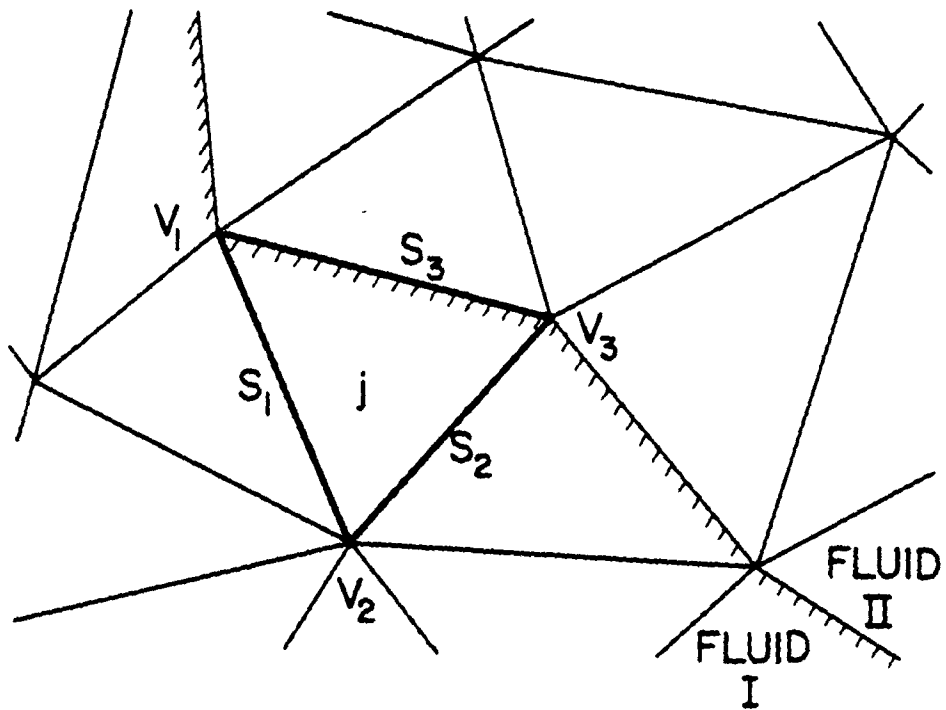
- 8) Modelling of Flows in Energetic Materials, E. Oran, Invited Presentation at ONR/AFOSR Workshop on Energetic Materials, Chestertown, MD, August, 1983
- 9) Numerical Simulation of Combustion Processes, E. Oran, Invited Colloquium at Naval Surface Weapons Center, April, 1983
- 10) Fluid Dynamics in Reacting Systems, E. Oran, Invited colloquium at the University of Paris, June, 1983.

REFERENCES

- Boris, J.P., (1979) ADINC: An Implicit Lagrangian Hydrodynamics Code, NRL Memorandum Report 4022, Naval Research Laboratory, Washington, D.C. 20375.
- Clift, R., J. A. Grace and M. E. Weber, (1978) Bubbles, Drops and Particles (Academic Press, N.Y., 1978).
- deBoor, C., (1978) A Practical Guide to Splines (Springer-Verlag, N.Y. 1978), pp. 316-322.
- Emery, M. H., M. J. Fritts and R. C. Shockley, (1981) Lagrangian Simulation of Taylor-Couette Flow, NRL Memorandum Report 4569, Naval Research Laboratory, Washington, D.C., 20375.
- Faeth, G. M., (1977) Current Status of Droplet and Liquid Combustion, Prog. Energy Comb. Sci. 3, 191.
- Faeth, G. M., (1983) Evaporation and Combustion of Sprays, Prog. Energy Comb. Sci., to be published.
- Foote, G. B., (1973) A Numerical Method for Studying Liquid Drop Behavior; Simple Oscillations, J. Comp. Phys. 11, 1973, 507-530.
- Fritts, M. J., (1976) A Numerical Study of Free-Surface Waves, SAI Report SAI-76-528-WA, Science Applications Incorporated, McLean, Va.
- Fritts, M. J., (1976a) Lagrangian Simulations of the Kelvin-Helmholtz Instability, SAI Report SAI-76-632-WA, Science Applications Incorporated, McLean, Va.
- Fritts, M. J. and Boris, J. P., (1979) The Lagrangian Solution of Transient Problems in Hydrodynamics using a Triangular Grid, J. Comp Phys. 31, 173-215.

- Fritts, M. J., F. W. Miner and O. M. Griffin, (1980) Numerical Calculation of Wave-Structure Interactions, Computer Methods in Fluids (Pentech Press, 1980), Chap. 1.
- Fritts, M. J., Oran, E. S. and Boris, J. P., (1981) Lagrangian Fluid Dynamics for Combustion Modelling, NRL Memorandum Report 4570, Naval Research Laboratory, Washington, D.C. 20375.
- Fritts, M. J., Fyfe, D. E., and Oran, E. S., (1982) Numerical Simulations of Droplet Flows with Surface Tension, Proceedings of the 1982 Winter Meeting of the American Society of Mechanical Engineers, Phoenix, Arizona.
- Jones, W. W. and Boris, J. P., (1979) Flame, A Slow-Flow Combustion Model, NRL Memorandum Report 3970, Naval Research Laboratory, Washington, D.C. 20375.
- Landau, L. D. and Lifshitz, L. D., (1975) Fluid Mechanics, Pergamon Press, New York, p. 230-234.
- Oran, E. S. and Boris, J. P., (1981) Detailed Modelling of Combustion Systems, Prog. Energy Comb. Sci. 7, 1-71.
- Rayleigh, Lord, (1879) On the Capillary Phenomena of Jets, Proc. Roy. Soc. (London) 29, 71-97.
- Williams, A., (1973) Combustion of Droplets of Liquid Fuels, Comb. Flame 21, 1-31.

a)



b)

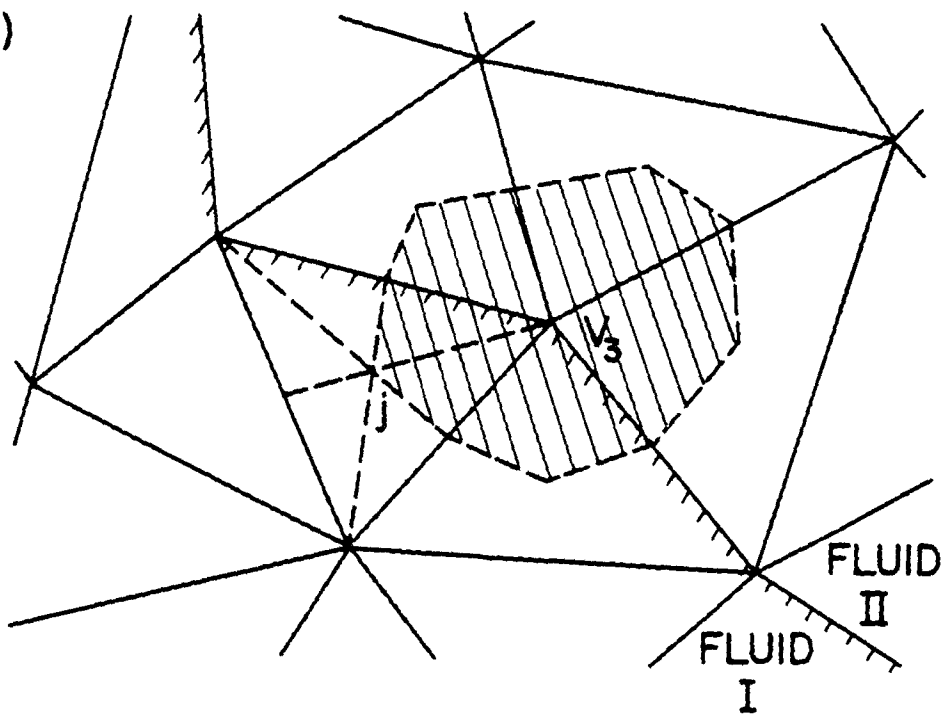


Fig. 1 A section of a triangular mesh including an interface and showing a vertex cell.

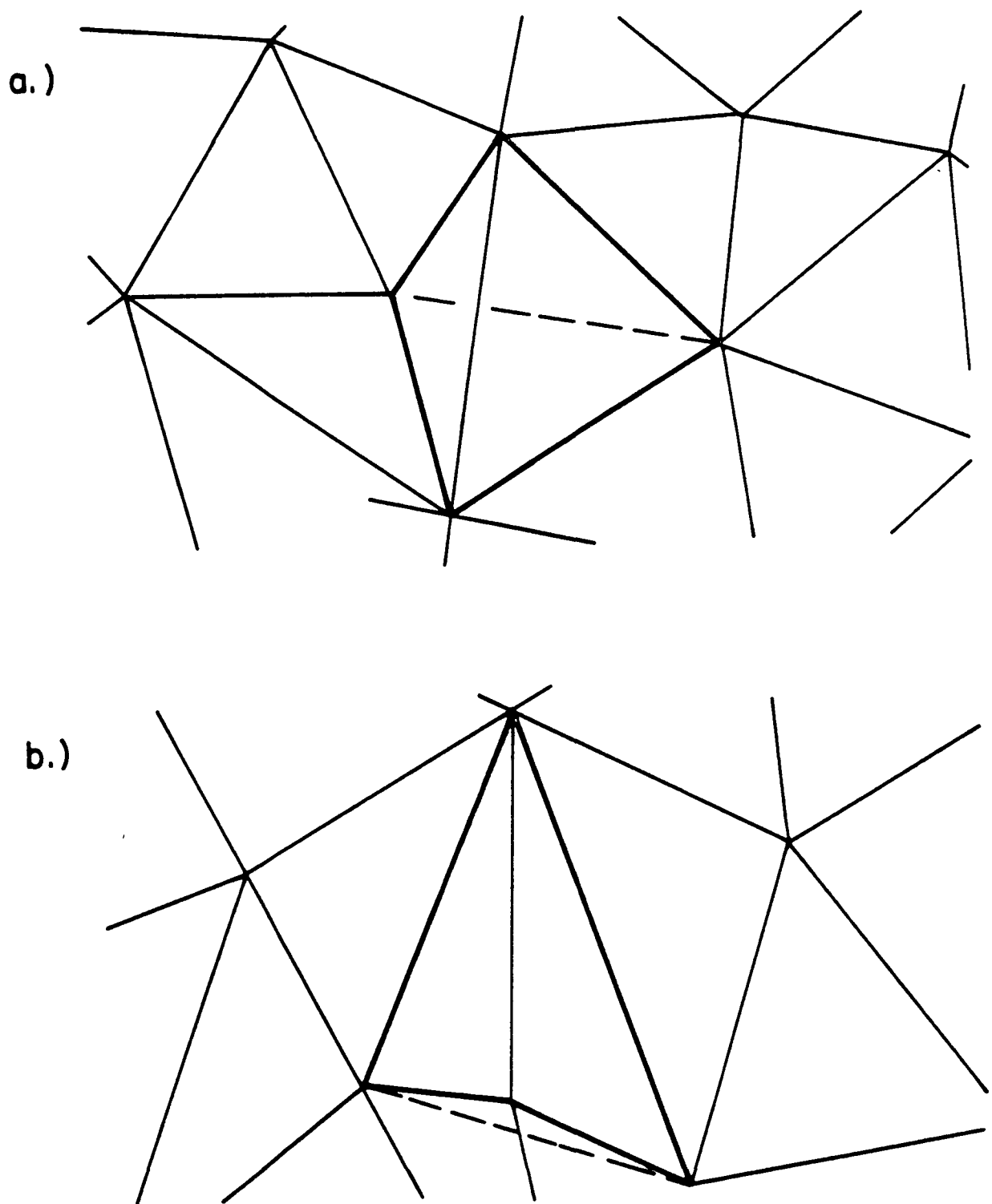


Fig. 2. Portions of a grid illustrating possible reconnections. In part (a) the dashed diagonal will be chosen for the shaded quadrilateral rather than the present, longer, diagonal. In part (b) the diagonal cannot be reconnected since the alternative diagonal, though shorter, lies outside the "inverted" quadrilateral.

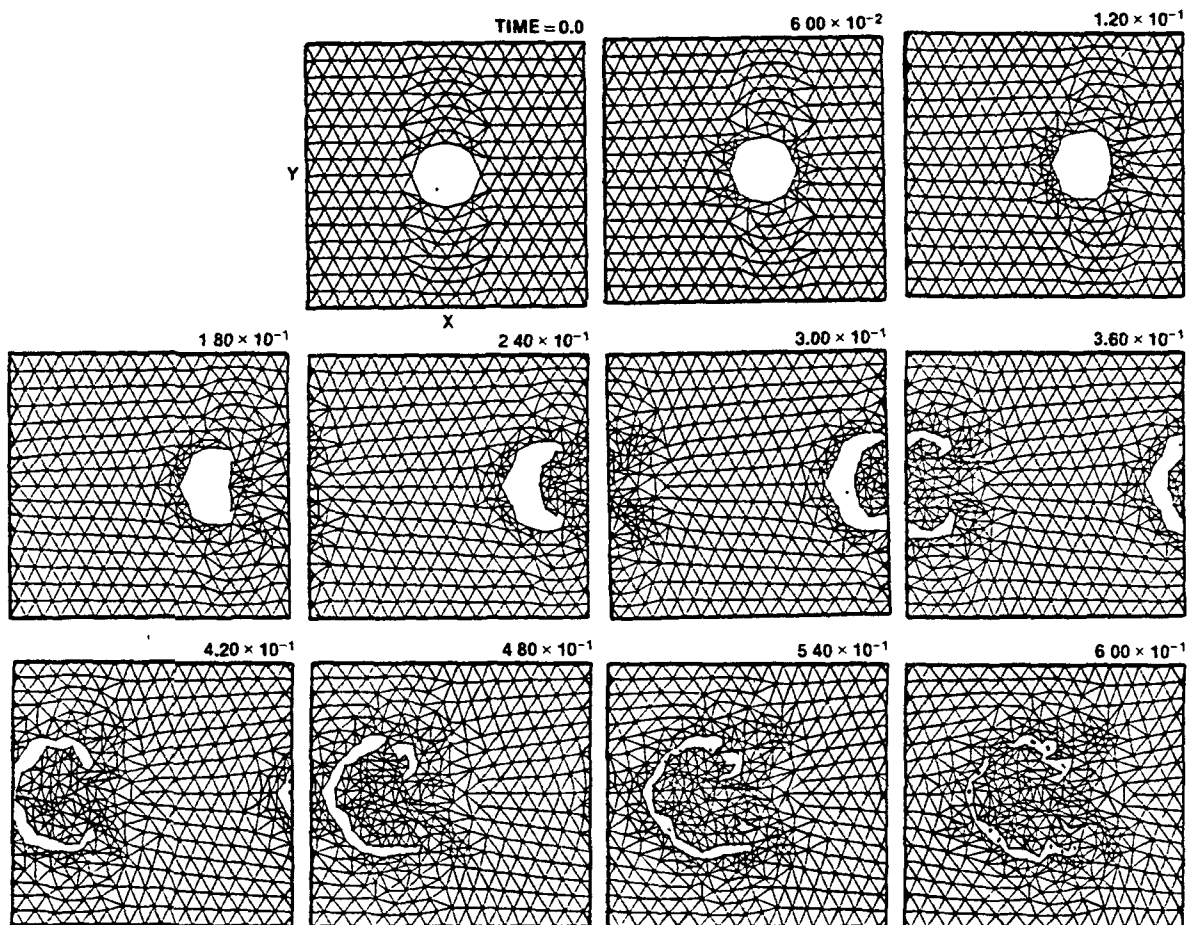


Fig. 3 Frames from a computer generated movie of a simulation without surface tension or viscosity showing droplet deformation and shattering due to the external flow.

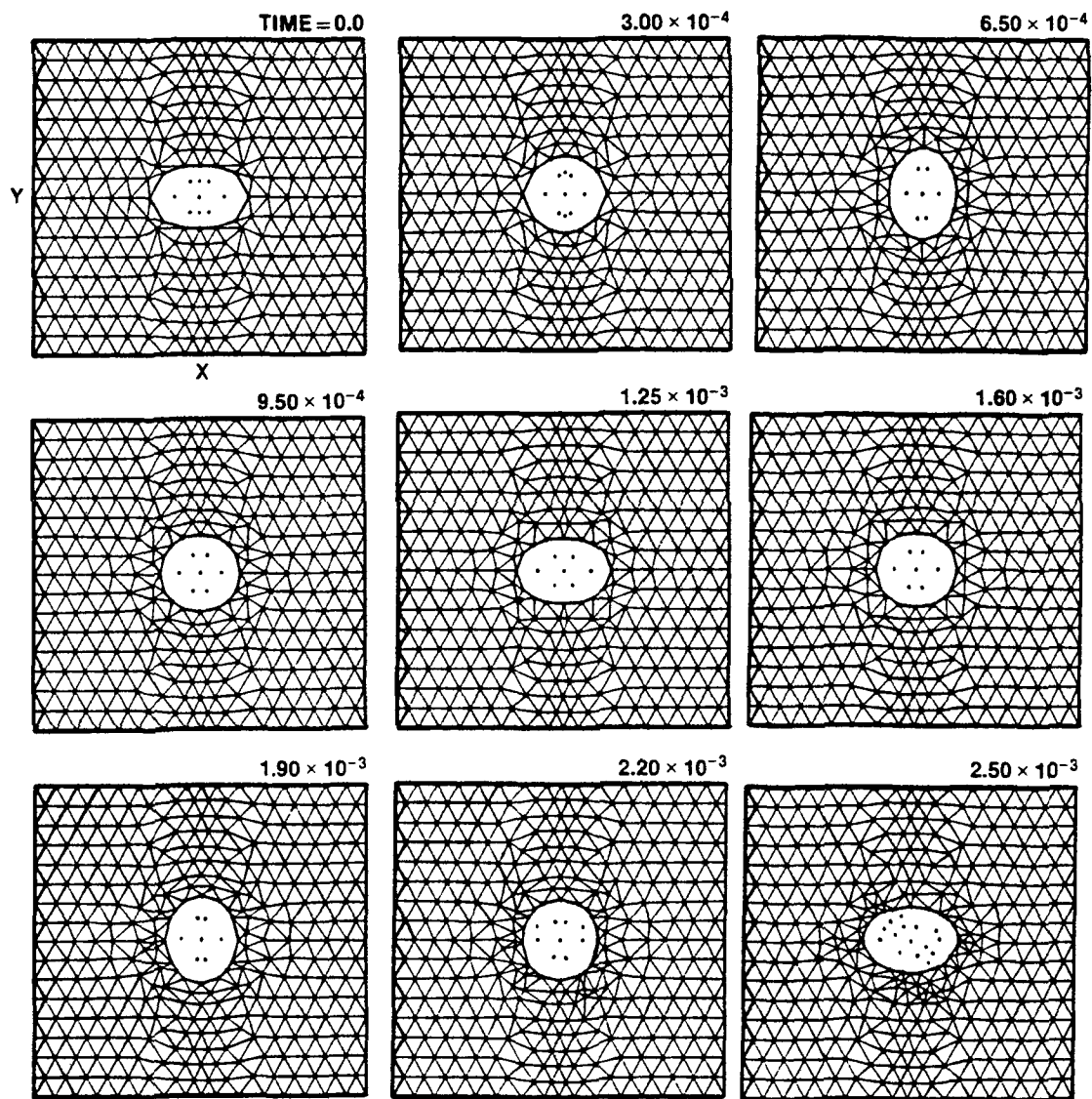


Fig. 4 Computer generated frames from a Lagrangian simulation of an oscillating droplet for two periods of an $\eta = 2$ normal mode.

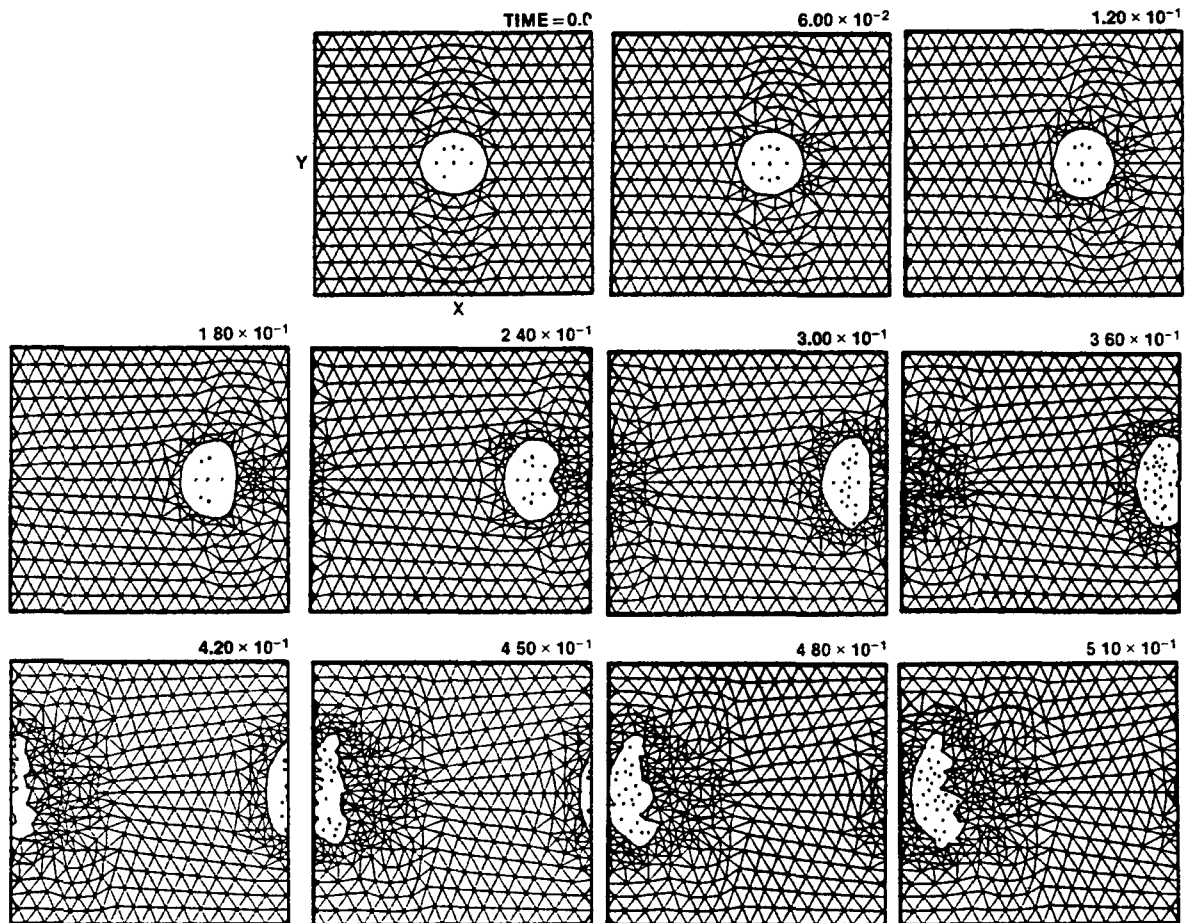


Fig. 5 Frames from a simulation with surface tension but no viscosity showing droplet deformation due to the external flow.

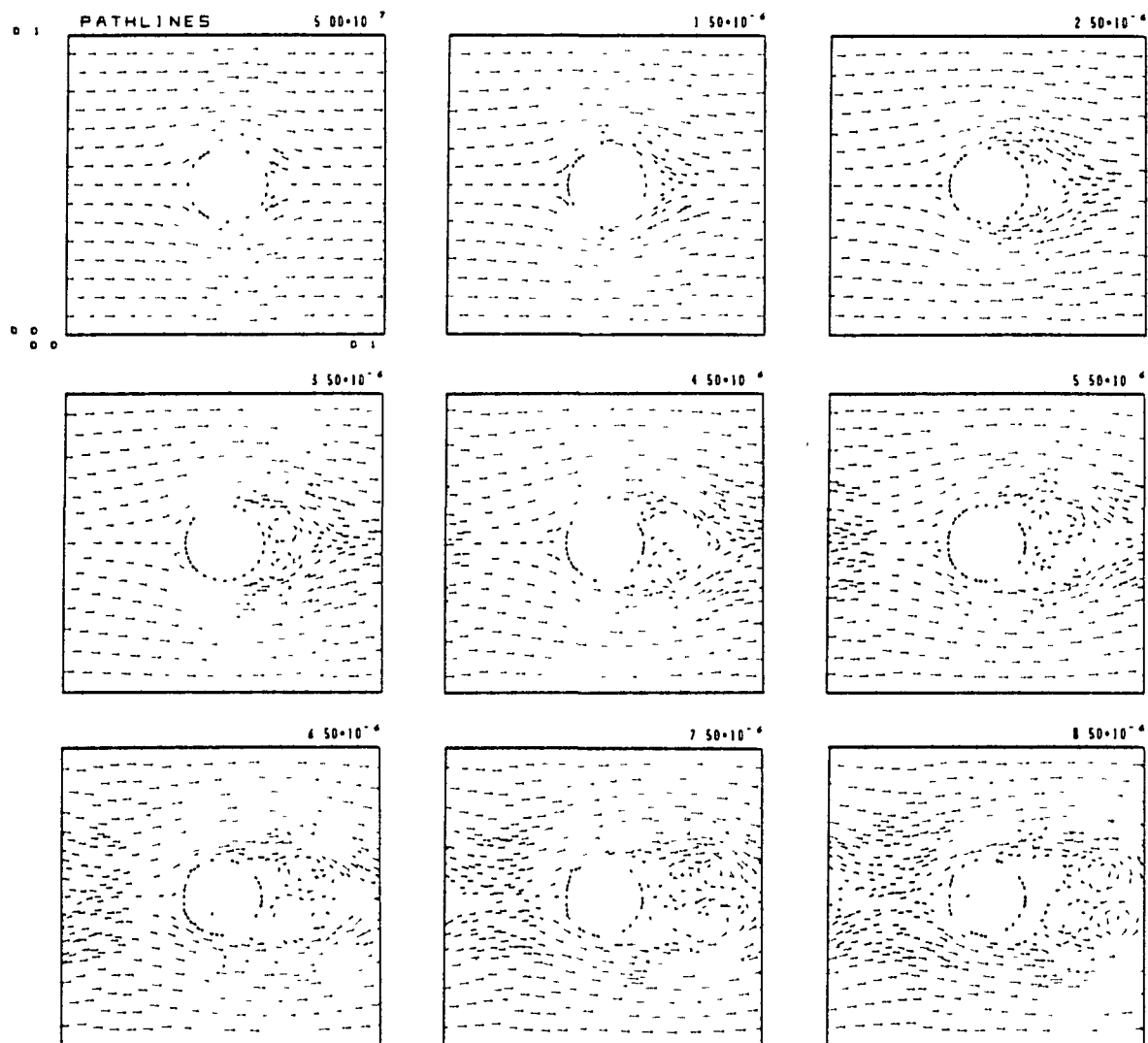


Fig. 6 Pathlines of the fluid flow from a computer generated movie of incompressible air flow past a deforming kerosene droplet. Heads of the pathlines are the current vertex positions and the tails are made up of the previous six positions. The flow speed is 100 m/sec. and the effects of surface tension and viscosity are included.

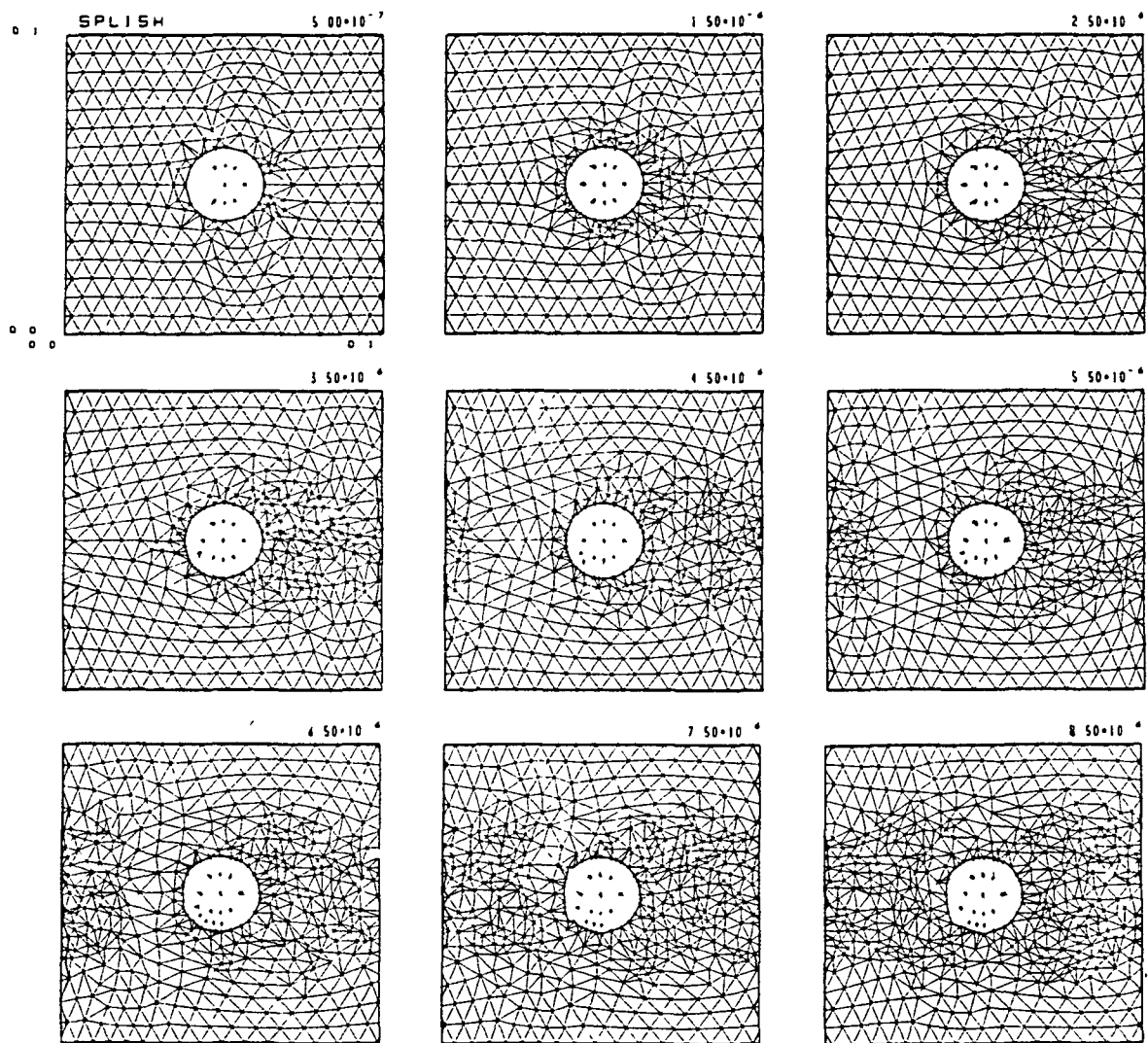


Fig. 7 Frames showing the triangular grid at the same times as shown for the pathlines in Figure 6.

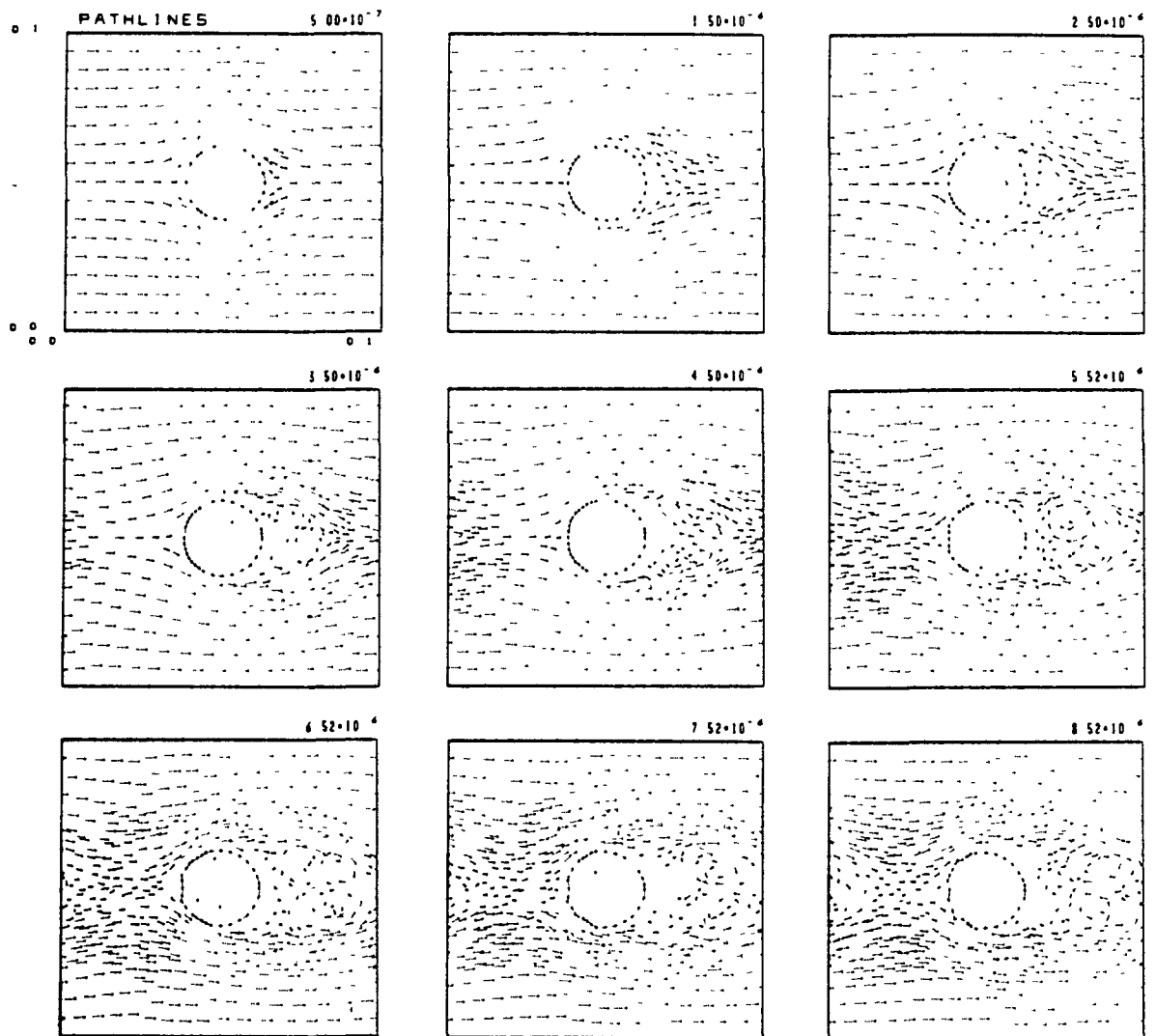


Fig. 8 Pathlines for the fluid flow when the flow speed is 120 m/sec. All other parameters for the calculation are identical with those for the simulation shown in Figure 6.

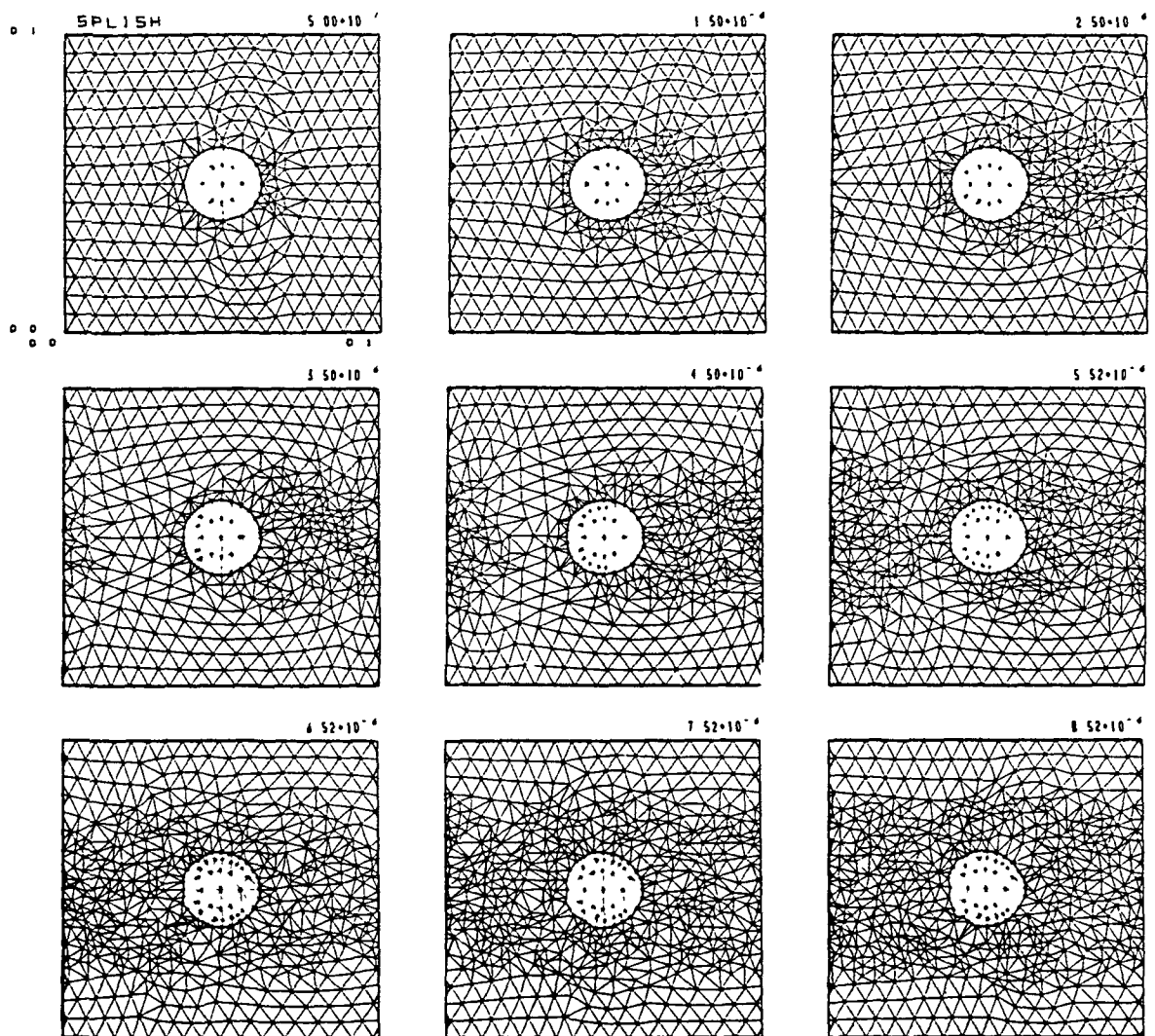


Fig. 9 The restructuring triangular grid at the times corresponding to the frames in Figure 8.

NASA Contractor Report NASA CR-168263

Sound Film Supplement C312
Numerical Simulations of Fuel Droplet Flow
Using a Lagrangian Triangular Mesh

Film Supplement to NASA Contractor Report CR-168263 by M. J. Fritts, D. E. Fyfe and E. S. Oran (1983). Computer generated movies show computations of droplet breakup and oscillations. The interface is tracked using color graphics. A Lagrangian triangular grid system is used showing the droplet distortion and breakup with resultant reconnection of points in the space.

12 minutes, color, sound.

Film supplement C312 is available on request to:

Chief, Publications Division
National Aeronautics and Space Administration
Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

Cut

Date _____	
Please send on loan Film Supplement to report NASA-CR 168263. Lewis File C-312.	
Name of organization _____	
Street number _____	
City and State _____	Zip code _____
Attention: Mr. _____	
Title _____	

Place
Stamp
Here

National Aeronautics and Space Administration
Lewis Research Center
Attention: Chief, Publications Division (60-1)
21000 Brookpark Road
Cleveland, OH 44135